Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/IL05/000319

International filing date: 21 March 2005 (21.03.2005)

Document type: Certified copy of priority document

Document details: Country/Office: US
Number: 60/555,717
Filing date: 23 March 2004 (23.03.2004)

Date of receipt at the International Bureau: 29 April 2005 (29.04.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

PA 1284555

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

February 17, 2005

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: *60/555,717*
FILING DATE: *March 23, 2004*

By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

N. Woodson

N. WOODSON
Certifying Officer

Please type a plus sign (+) inside this box ——→ [+]

PTO/SB/16 (5-03)
Approved for use through 04/30/2003. OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
der the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

INVENTOR(S)

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Hillel	SOLOW	Beit Shemesh, Israel
Anatoly	SELDIN	Jerusalem, Israel
David	MAIL	Zur Igal, Israel

□ *Additional inventors are being named on the_____ separately numbered sheets attached hereto*

TITLE OF THE INVENTION (280 characters max)

DIGITAL RIGHTS MANAGEMENT SYSTEM FOR MULTIMEDIA MESSAGES PROTECTION AND TRACKING

Direct all correspondence to: **CORRESPONDENCE ADDRESS**

□ Customer Number [ ] ——→ *Place Customer Number Bar Code Label here*

OR *Type Customer Number here*

☒ Firm or Individual Name	L. Friedman				
Address	Welsh & Katz, Ltd.				
Address	120 S. Riverside Plaza, 22nd Floor				
City	Chicago	State	Illinois	ZIP	60606
Country	USA	Telephone	312-655-1500	Fax	312-655-1501

ENCLOSED APPLICATION PARTS *(check all that apply)*

☒ Specification *Number of Pages* 52
including 44 pg Appendix
☒ Drawing(s) *Number of Sheets* 3
☒ Application Data Sheet. See 37 CFR 1.76

□ CD(s), Number

□ Other (specify)

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT *(check one)*

☒ A check or money order is enclosed to cover the filing fees
☒ The Director is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number 23-0920
□ Payment by credit card. Form PTO-2038 is attached.

FILING FEE AMOUNT ($)
$160.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.
☒ No.
□ Yes, the name of the U.S. Government agency and the Government contract number are: ____________

Respectfully submitted,

Date 3/23/04

SIGNATURE

REGISTRATION NO. 37,135
(if appropriate)
Docket Number: 7251/91771

TYPED or PRINTED NAME L. Friedman

TELEPHONE 312-655-1500\

U.S. PTO
17271 60555717
032304

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is used by the public to file (and by the PTO to process) a provisional application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the complete provisional application to the PTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Mail Stop Provisional Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

P19LARGE/REV05

PTO/SB/17 (10-03)
Approved for use through 07/31/2006. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
der the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2004

Effective 10/01/2003. Patent fees are subject to annual revision.

□ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT	($) $160.00

Complete if Known

Application Number	
Filing Date	23 March 2004
First Named Inventor	SOLOW et al.
Examiner Name	
Art Unit	
Attorney Docket No.	7251/91771

METHOD OF PAYMENT (*check all that apply*)

☒ Check □ Credit card □ Money □ Other □ None

□ Deposit Account:

Deposit Account Number	23-0920
Deposit Account Name	Welsh & Katz, Ltd.

The Director is authorized to: *(check all that apply)*

□ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additonal fee(s) or any underpayment of fee(s)

□ Charge fee(s) indicated below, except for the filing fee

to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	770	2001	385	Utility filing fee	
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	160.00

SUBTOTAL (1) ($) $160.00

2. EXTRA CLAIM FEES FOR UTILITY AND

	Extra Claims				Fee from below		Fee Paid
Total Claims		-20** =	0	X		=	0.00
Independent Claims		-3** =	0	X		=	0.00
Multiple Dependent						=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple dependent claim, if not paid
1204	86	2204	43	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) $0.00

***or number previously paid, if greater; For Reissues, see above*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet	
1053	130	1053	130	Non - English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	2,010	2255	1,005	Extension for reply within fifth month	
1401	330	2401	165	Notice of Appeal	
1402	330	2402	165	Filing a brief in support of an appeal	
1403	290	2403	145	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,330	2453	665	Petition to revive - unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR § 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Statement	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	770	2809	385	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR § 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) ____________________

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($)

SUBMITTED BY Complete *(if applicable)*

Name *(Print/Type)*	L. Friedman	Registration No. *(Attorney/Agent)*	37,135	Telephone	312-655-1500
Signature				Date	23 March 2004

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Patentee: SOLOW, et al
Title: DIGITAL RIGHTS MANAGEMENT SYSTEM FOR MULTIMEDIA MESSAGES PROTECTION AND TRACKING
Serial No.:
Filing Date: 23 March 2004
Docket No. 7251/91771

Certificate of Express Mailing

Express Mail mailing label number EL 995603388 US

Date of Deposit: 23 March 2004

I hereby certify that this paper is being deposited with the United States Postal Service "Express Mail" Post Office to: Mail Stop Provisional Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450. This mailing includes Provisional Application Cover Sheet (1 pg); Fee Transmittal (1 pg) in duplicate; Check in the amount of $160.00; Specification (52 pgs, including 44 page Appendix) and Drawings (3 sheets); Application Data Sheet (3 pgs); and return receipt postcard.

The person mailing this paper is:

John Austin Rembert
Typed or Printed Name of Person Mailing Paper of Fee

John Austin Rembert
Signature of Person Mailing Paper or Fee

Application Data Sheet

Inventor Information

Inventor One Given Name::	Hillel
Family Name::	SOLOW
Postal Address Line One::	115 Shimon Street
City::	Beit Shemesh
Country::	Israel
Postal or Zip Code::	99543
Citizenship Country::	Israel

Inventor Two Given Name::	Anatoly
Family Name::	SELDIN
Postal Address Line One::	18/3 Rabbi Meir Street
City::	Jerusalem
Country::	Israel
Postal or Zip Code::	93185
Citizenship Country::	Israel

Inventor Three Given Name::	David
Family Name::	MAIL
Postal Address Line One::	16 Emek Dotan Street
City::	Zur Igal
Country::	Israel
Postal or Zip Code::	44862
Citizenship Country::	Israel

Correspondence Information

Name Line One::	Welsh & Katz, Ltd.
Name Line Two::	L. Friedman
Address Line One::	22nd Floor
Address Line Two::	120 South Riverside Plaza
City::	Chicago
State or Province::	IL
Postal or Zip Code::	60606
Telephone Number::	(312) 655-1500
Fax::	(312) 655-1501

Application Information

Title Line One::	DIGITAL RIGHTS MANAGEMENT SYSTEM FOR MULTIMEDIA MESSAGES
Title Line Two::	PROTECTION AND TRACKING
Total Drawing Sheets::	3
Application Type::	Provisional
Docket Number::	7251/91771

Representative Information

Registration Number One::	24,003
Registration Number Two::	22,839
Registration Number Three::	28,903
Registration Number Four::	27,429
Registration Number Five::	25,060
Registration Number Six::	22,053
Registration Number Seven::	27,466
Registration Number Eight::	29,434
Registration Number Nine::	29,054
Registration Number Ten::	29,381
Registration Number Eleven::	34,044
Registration Number Twelve::	27,600
Registration Number Thirteen::	34,137
Registration Number Fourteen:	38,110

Registration Number Fifteen::	39,724
Registration Number Sixteen:	39,021
Registration Number Seventeen:	37,963
Registration Number Eighteen:	37,135
Registration Number Nineteen:	40,604
Registration Number Twenty:	37,435
Registration Number Twenty-One:	45,195
Registration Number Twenty-Two:	40,687
Registration Number Twenty Three:	41,050

Assignee Information

Assignee Name: NDS Limited

Assignee Address: One London Road
Staines, Middlesex TW18 4EX
United Kingdom

P-162prv3b.doc P-162 DM/DZ 23MAR04/3

DIGITAL RIGHTS MANAGEMENT SYSTEM FOR MULTIMEDIA MESSAGES PROTECTION AND TRACKING

FIELD OF THE INVENTION

The present invention relates to content protection for multimedia content.

BACKGROUND OF THE INVENTION

Background technologies relevant to understanding the present invention include:

- Simple Object Access Protocol (SOAP), which is described on the World Wide Web at: www.w3.org/2000/xp/Group/

- SOAP Security Extensions: Digital Signature (SOAP DSIG), which is described on the World Wide Web at: www.w3.org/TR/SOAP-dsig/

The disclosures of all references mentioned above and throughout the present specification (including, without limitation, references mentioned in Appendix A), as well as the disclosures of all references mentioned in those references, are hereby incorporated herein by reference.

SUMMARY OF THE INVENTION

The present invention, in preferred embodiments thereof, provides a system and method for content protection for multimedia content distributed in a mobile network by Multimedia Message Service Center (MMSC). In preferred embodiments, the system includes a Digital Rights Management (DRM) server connected to the MMSC by a dedicated DRM protocol or by connection via selected MMSC protocols. The system may also include a DRM User Agent (UA) on mobile handsets.

BRIEF DESCRIPTION OF THE DRAWINGS AND APPENDIX

The present invention will be understood and appreciated more fully from the following detailed description, taken in conjunction with the drawings in which:

Fig. 1 is a simplified block diagram illustration of a 3GPP MMSC System having an MM9 interface;

Fig. 2 is a simplified block diagram illustration of a Multimedia Message Service Center (MMSC) including a Digital Rights Management (DRM) server, constructed and operative in accordance with a preferred embodiment of the present invention; and

Fig. 3 is a simplified block diagram illustration of a Multimedia Message Service Center (MMSC) including a Digital Rights Management (DRM) server, constructed and operative in accordance with another preferred embodiment of the present invention.

The following appendix will aid in understanding the detailed description:

Appendix A, which is a particularly detailed description of one preferred implementation of an interface between a service provider and a DRM Server.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Reference is now made to Fig. 1, which is a 3GPP MMSC System diagram with an additional MM9 interface. Fig. 1 is based on the standard MMSC architecture as it is defined by 3GPP in reference [1] (a copy of which may be found on the World Wide Web at: webapp.etsi.org/action%5CPU/20040120/ts_123140v050900p.pdf) with the addition of a transcoder node connected by MM9 protocol.

It is appreciated that MM9 is not standardized yet and its implementation may be proprietary for each MMSC and transcoder vendor.

The DRM System for Multimedia Message (MM) protection preferably comprises, at the server side, one of the following:

I)

A DRM component, integrated within the MMSC through a dedicated protocol, similar to the current approach to MMSC - transcoder interconnection, as shown in Fig. 2. The dedicated protocol could be very similar to that described in Appendix A, with additional support for extended business models if desired. (It is appreciated that Appendix A comprises a particularly detailed description of one preferred implementation of an interface between a service provider and a DRM Server; the example of Appendix A is not meant to be limiting).

II) Stand-alone server(s) acting like a network probe and/or proxy, as shown in Fig. 3.

The system of Fig. 3 preferably protects / tracks content following the steps below. The DRM Server preferably:

1. Listens to MM7 protocol analyzing *MM7_submit.REQ* messages.
2. For each message, extracts from the message Value Added Service Provider (VASP) ID and Value Added Service (VAS) ID and compares them with a pre-defined list of the content vendors IDs that request content protection/tracking. If the content protection/tracking was required, then:
3. Generates hash data which uniquely identifies MM content and stores this data in the DRM DB together with other data about the message, such as, for example VASP & VAS IDs, message recipient(s) address(es)/distribution list(s), submission timestamp, subject, service code, message class, message distribution indicator and other appropriate parameters. Data hash technology is well-known in the art; any appropriate data hash technology may be used.
4. Optionally, analyzes content and replaces it by the same content with a watermark. This could be used with or instead the content hashing as described in step 3 immediately above
5. Analyzes every message transferred via proprietary MMSC-transcoder protocol (MM9). For messages, passed from MMSC to the transcoder, the DRM Server generates content hash and/or looks for content watermark. The generated hash is

compared with the stored hashes (or the found watermark compared with DB stored watermarks). If the same hash/watermark is found in the DRM DB, the incoming (to transcoder) message is marked as "to be protected/tracked" by storing incoming message parameters such as transaction ID, message ID or similar. Persons skilled in the art will appreciate that specific implementation details depend on the MMSC-transcoder protocol details, which differs for various vendors.

6. Verifies the same parameters (transaction ID, message ID or similar) for any outgoing (from transcoder) message while listening to MM9 protocol. If the message should be protected, the DRM Server encrypts content, fully or partially. Every attachment can be protected individually by using different algorithms and/or keys or all the attachments can be encrypted together with the single key. Any appropriate encryption algorithms and key management and key delivery mechanisms may be used.
7. Generates an additional post-transcoding hash data for the transcoded content and stores this data in the DRM DB.

Then the message is delivered by MMSC, preferably using standard methods known in the art, to the recipient MMS UA.

In order to implement the protection for the super distribution (i.e. forwarding all the content items received within the MM or selected content items only from one mobile subscriber to the another one) in case of Multimedia Message (MM) submitting (which may occur when only selected attachments of received MM are forwarded) and MM forwarding (entire MM), the DRM server preferably:

1. Listens to MM1 protocol and analyzes every *MM1_submit.REQ* message by generating a content hash. The hash is compared to the "post-transcoding" hashes in the DRM DB. If the generated hash matches a hash in the DB:
2. Registers message parameters such as originator, recipients list, timestamp and others as appropriate.
3. Verifies the rights of the recipient(s) to receive the message. If the recipient(s) is not allowed to receive the message, e.g. the MM was requested to be forward-locked by the content or service provider, the DRM Server may:
    a. Silently ignore the MM forward attempt.
    b. Inform the MM originator that the delivery is not allowed, by SMS, MMS, WAP push or other allowed methods.
    c. Inform MM recipient that there was a MM forwarding attempt.

   If the message delivery to the receipient(s) is allowed, optionally the DRM Sever may request the recipient to purchase the appropriated rights for the forwarded content first and execute all the steps below only when the purchase confirmation will be received. If the DRM Server is allowed to deliver the message:
4. Decrypts content and replace the encrypted attachments by clear content inside the *MM1_submit.REQ* message. Then the message is delivered to the MMSC Server/Relay which in turn may pass it to the transcoder. Further operations are defined in steps (5)-(7) for MM9 above.

5. In addition, listens to MM1 protocol and analyzes every *MM1_retrieve.REQ* and *MM1_retrieve.REQ* messages, extracting message reference, typically in the form of a Universal Resource Identifier (URI), which URIs are well-known in the art, from *MM1_retrieve.REQ* and storing the message reference in the DRM DB if content hash generated by the DRM Server for content passed in *MM1_retrieve.RES* matches with a hash stored in the DRM DB.
6. Listens to *MM1_forward.REQ*, extracting message URI and comparing with the stored URIs. For the matching URIs, registers in the DRM DB the recipient's address and originator address presented in *MM1_forward.REQ* message.

Additionally, persons skilled in the art will appreciate that the DRM server preferably, in both cases (I and II), provides portal functionality for the DRM UA to supply keys as necessary, both on DRM UA request or provided by DRM Server by an appropriate push method such as SMS, WAP push or MMS.

At the handset side the DRM User Agent (UA) implementation can be one of the following:

1) No DRM UA
In this case the DRM Server alone preferably supports the following business models: content tracking, network forward lock (with/without notification of the originator), controlled super distribution.
2) DRM UA built-in the MMS UA.
3) DRM UA acting as a stand-alone application / group of applications (such as, for example, a Symbian recognizer [which is well-known in the art and described on the World Wide Web at: www.symbian.com/developer/techlib/v70sdocs/doc_source/devguides/cpp/applicationframework/recognizersoverview.guide.html] + UA) or as a part of a third-party stand-alone application (DRM UA for a player or browser), when the application is associated with the protected content types (and called by the Operating System (OS) when a MMS with protected content is received)
4) DRM UA acting as a part of the native phone software (OS, drivers etc) (including any HW implementation) – in the most common case is the same as (1); other options: part of a web/WAP browser..

For cases (2)-(4) DRM Server and DRM UA will support any appropriate business model implemented in the given DRM System, e.g. pay-per-count, pay-per-time, rental or permanent purchase.

Reference:
[1] ETSI TS 123 140 V5.9.0 (2003-12)
Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunication System (UMTS);
Multimedia Messaging Service (MMS);
Functional description;
Stage 2

(3GPP TS 23.140 version 5.9.0 Release 5)

It is appreciated that various features of the invention which are, for clarity, described in the contexts of separate embodiments may also be provided in combination in a single embodiment. Conversely, various features of the invention which are, for brevity, described in the context of a single embodiment may also be provided separately or in any suitable subcombination.

It will be appreciated by persons skilled in the art that the present invention is not limited by what has been particularly shown and described hereinabove. Rather the scope of the invention is defined only by the claims which follow:

What is claimed is:

CLAIMS

1. Apparatus substantially as described hereinabove.

2. Apparatus substantially as shown in the drawings.

3. A method substantially as described hereinabove.

4. A method substantially as shown in the drawings.

5. A system substantially as described hereinabove.

6. A system substantially as shown in the drawings.

FIG. 1

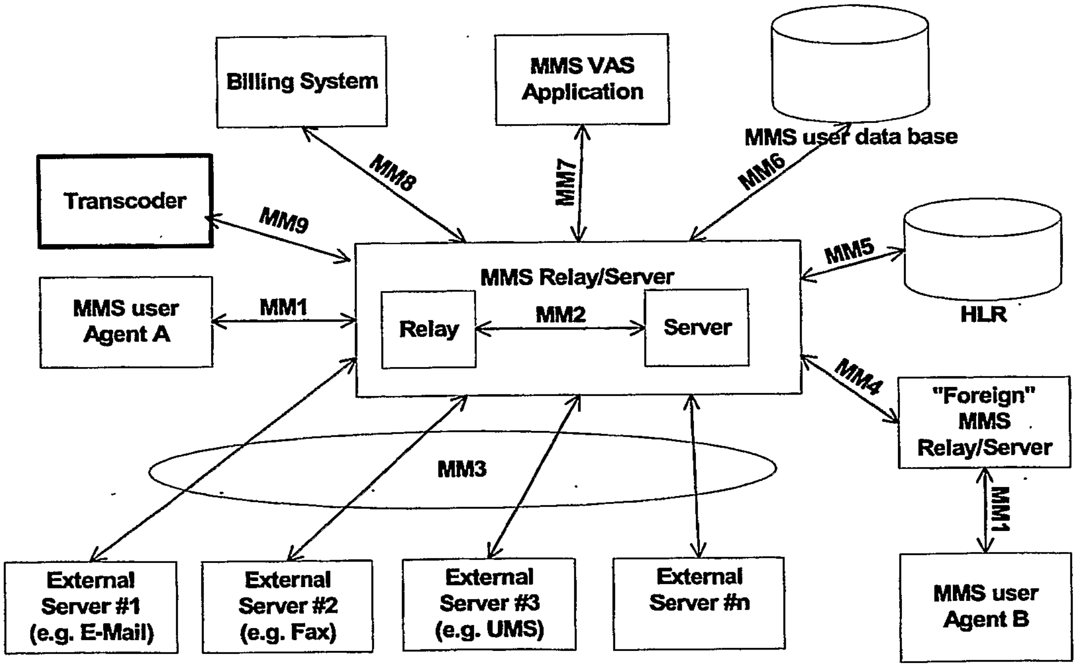

Billing System
MMS VAS Application
MMS user data base
Transcoder
MM9
MM8
MM7
MM6
MMS Relay/Server
MM5
HLR
MMS user Agent A
MM1
Relay
MM2
Server
MM4
"Foreign" MMS Relay/Server
MM3
MM1
External Server #1 (e.g. E-Mail)
External Server #2 (e.g. Fax)
External Server #3 (e.g. UMS)
External Server #n
MMS user Agent B

FIG. 2

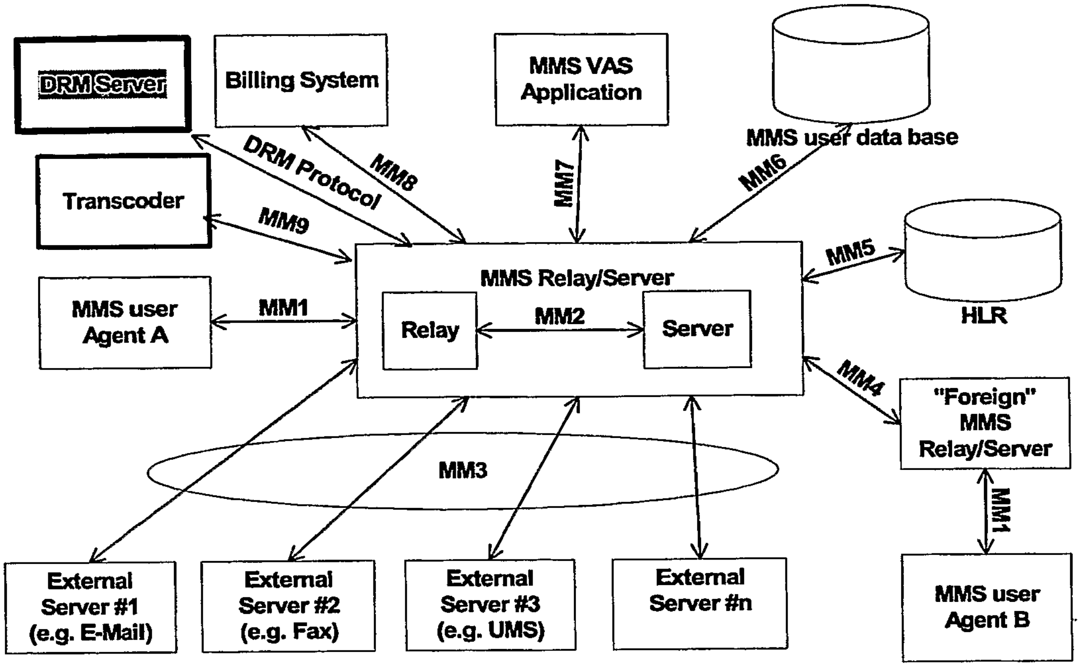
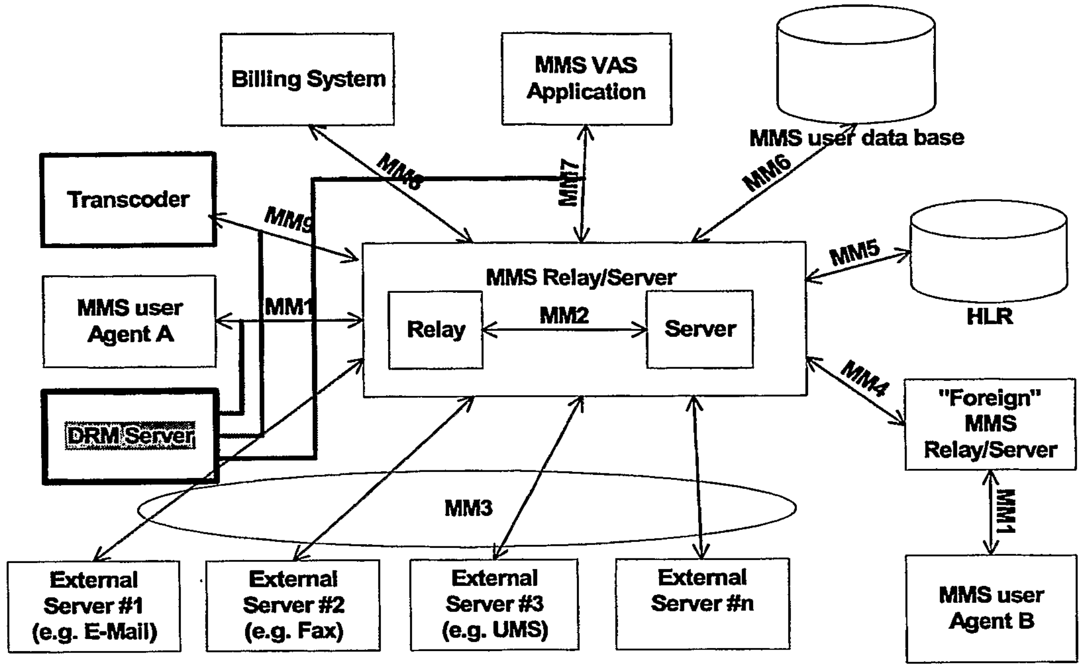

DRM Server
Billing System
MMS VAS Application
MMS user data base
Transcoder
DRM Protocol
MM8
MM7
MM6
MM9
MMS Relay/Server
MM5
HLR
MMS user Agent A
MM1
Relay
MM2
Server
MM4
"Foreign" MMS Relay/Server
MM3
MM1
External Server #1 (e.g. E-Mail)
External Server #2 (e.g. Fax)
External Server #3 (e.g. UMS)
External Server #n
MMS user Agent B

FIG. 3


Billing System
MMS VAS Application
MMS user data base
Transcoder
MM8
MM7
MM6
MM9
MMS Relay/Server
MM5
HLR
MMS user Agent A
MM1
Relay
MM2
Server
MM4
"Foreign" MMS Relay/Server
DRM Server
MM3
MM1
External Server #1 (e.g. E-Mail)
External Server #2 (e.g. Fax)
External Server #3 (e.g. UMS)
External Server #n
MMS user Agent B

APPENDIX A

DRM Server Service Provider API

Table of Contents

List of Tables

List of Figures

1. Scope

This document describes the interface between the Service Provider (SP) and NDS Digital Rights Management (DRM) server.

2 Related Documents

Doc. Designation	Document Title
[1] OMA-Download DRMCF-v1_0-20031113-C:	OMA DCF file format
[2] OMA-Download-DRM-V1_0-20031031-C:	OMA DRM
[3] OMA-Download-DRMREL-V1_0-20031031-C	OMA DRM Rights expression language

The OMA reference documents listed above can be found at the following URL:

http://www.openmobilealliance.org/release_program/enabler_releases.html

3 Overview

This interface defines communication between the service provider and NDS DRM server. The main functions of the interface are:

- Protecting content
- Generating a Rights Object (RO) to regulate content rendering by a Mobile Subscriber (mobile subscriber)

The interface supports all types of content protection (Forward Lock, Combine Delivery, Separate Delivery) and full functionality of ROs defined in [3]. RO functionalities are triggered by a mobile subscriber request to a service provider. The structure of the mobile subscriber request is outside the scope of this specification.

Additional functionalities required for the DRM server but not included in the interface definition below are:

- Service provider registration/authentication
- Mobile subscriber registration/authentication

The SP registration is triggered by the Management Station (MGMS). The MGMS shell registers each SP and SP service in the DRM server. The SP registration protocol, as well as SP authentication, is outside the scope of this specification.

Mobile subscriber authentication is performed by the SP upon a mobile subscriber's request for content. The registration/authentication can include the mobile subscriber identity, e.g., MSISDN, XID, cookies as well as the handset type (e.g. Nokia 6220) used by mobile subscriber.

3.1 DRM System Component Interaction

Figure 1 below describes how the DRM system components interact.

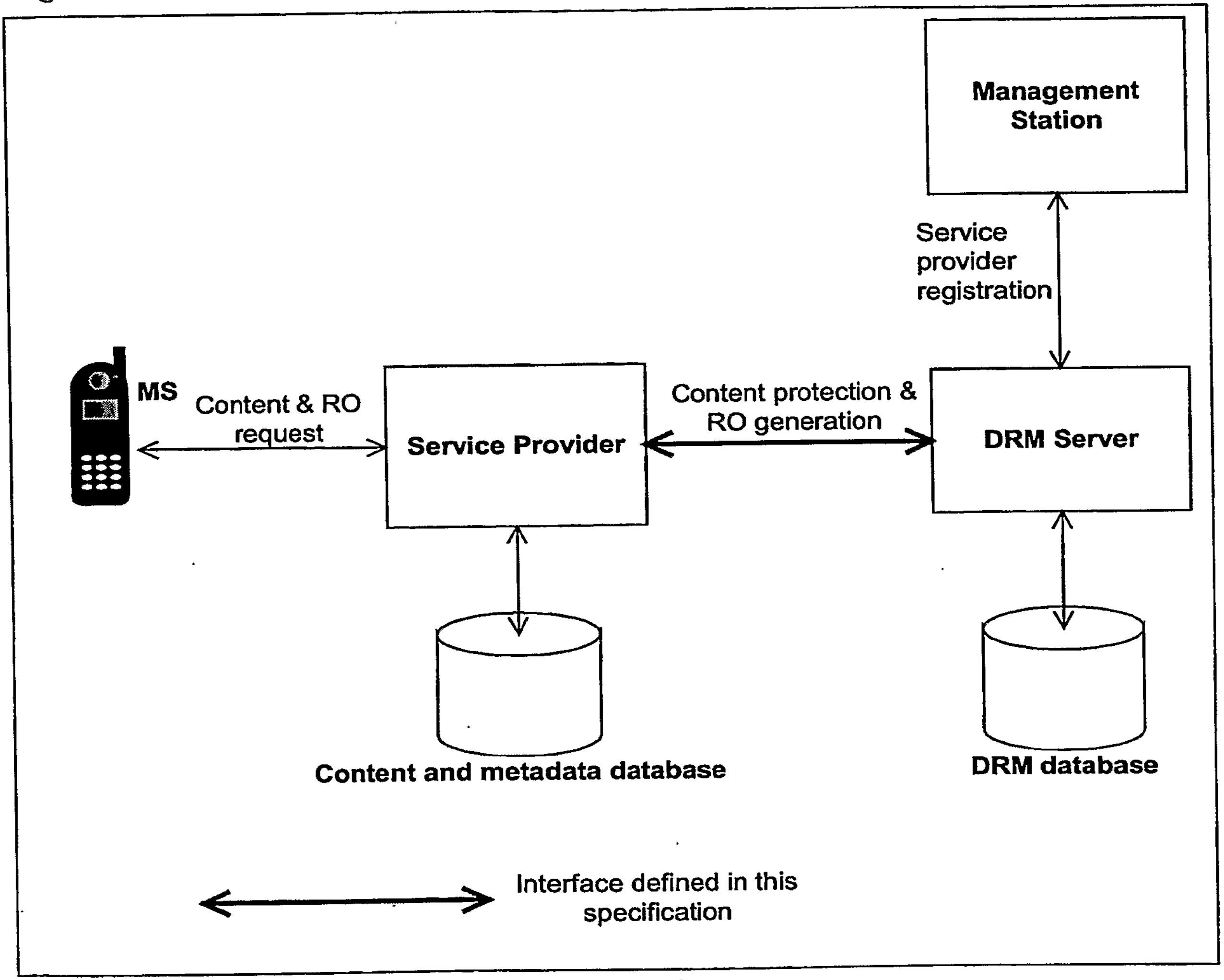

Management Station
Service provider registration
MS
Content & RO request
Service Provider
Content protection & RO generation
DRM Server
Content and metadata database
DRM database
Interface defined in this specification

Figure 1: Overview. DRM System Component Interaction

3.2 DRM Server's Role

In the scope of this document, the DRM server protects content and generates the RO for the content consumption by the mobile subscriber. The DRM server role is to optimally protect content for each mobile subscriber's handset, providing each handset with the best security level it can accept.

For example: if a specific handset supports the Forward Lock protection method only, the content for this handset should be Forward Lock-protected. If another handset supports both Forward Lock and Separate Delivery protection methods, the content for this handset should be protected by the Separate Delivery method and an appropriate RO should be generated and delivered to the handset as defined in specifications [2], [3].

The content protection method is chosen by the DRM server according to the content's handset type and MIME type. Choosing a content protection method is transparent to the SP.

4. Interface Principles

The interface described in this document uses Simple Object Access Protocol (SOAP) for the messages exchange.

4.1 Service Provider Authentication

SP authentication is performed by the DRM server during the communication session with the SP. The authentication details are outside the scope of this document.

4.2 Mobile Subscriber Registration/Authentication

Mobile subscriber authentication (both Mobile Subscriber ID and Mobile Subscriber type) is performed by the SP. The SP is responsible for transferring the Mobile Subscriber ID and handset type to the DRM server.

4.3 Permissions Handling

The SP must provide the DRM server with all information needed for content protection and RO generation. One of the parameters necessary for RO generation is permission as defined in [3]. Permissions are the part of the business scenario associated with the content, e.g., how often can content be viewed. The business scenario defines how content can be bought by the mobile subscriber.. The sequence for implementing permissions for a mobile subscriber is as follows:

1. The mobile subscriber selects the content.
2. The mobile subscriber purchases rights for the content.
3. The SP sends the permissions associated with these rights and the handset model name to the DRM server.
4. The DRM server checks the handset type to determine whether permissions can be implemented.

For example, if the handset supports only Forward Lock protection (Nokia 6600), then permissions such as "play 5 times" cannot be implemented. If the handset is able to implement all permissions, the content will be optimally protected by the handset and an RO will be generated if needed. If the handset is not able to implement certain permissions, the DRM server will implement the best content protection available. Notification, with an error code, will be sent to the SP.

Optionally, an SP can request from the DRM server a list of permissions that the particular handset type can implement for a particular asset type. This information can be used by the SP to display to the mobile subscriber only the permissions that the handset can implement.

4.4 Security Considerations

The SOAP digital signature (SOAP DSIG) can be used to secure communication between the SP and the DRM server.

4.5 Real Time Asset Protection

The DRM server supports both real time and pre-encrypted asset protection.

5 Interface Structure

The interface structure supports SOAP communication, where each transaction consists of request and response XML messages transferred over HTTP.

1. Protect content request
   a. RT-ProtectContent-reqRequest
   b. ProtectContentResponse
2. RO request
   a. RO-reqRequest
   b. RO-Response
3. DRM permission request
   a. DRM-permission-reqRequest
   b. DRM-permission-Response

6 XML Parameter Description

Table 1 below describes the interface parameters.

Table 1: Parameter Description

Parameter Name	Description
Version	This parameter defines the interface version.
SrvId	This parameter uniquely identifies the service in the DRM server.
BinaryRO	RO in WBXML format.
cid	Unique identifier generated by DRM server per each content encryption. Is defined in the DCF header [1].
MobSubsId	Mobile subscriber ID. This complex parameter uniquely identifies the subscriber in the network.
MobSubsIdType	Sub-parameter of MobSubsId. This parameter defines which type of identification is used by the SP.
MobSubsIdValue	Sub-parameter of MobSubsId. This parameter defines the ID value used, according to the chosen type.
MS_model	Mobile subscriber handset model
TsourceContent	This complex parameter incorporates all source (clear) content parameters used by the DRM server to protect content.
SourceContentID	Sub-parameter of tSourceContent. Unique ID of the clear content per SP This parameter is managed by SP and provided to DRM server as part of the RtProtectContentReq request.
ContentName	Sub-parameter of tSourceContent. This parameter is used in the DCF header, see [1].

Parameter Name	Description
ContentDescription	Sub-parameter of tSourceContent. The parameter is optional. This parameter is used in the DCF header, see [1].
MIME_type	Sub-parameter of tSourceContent. This parameter defines the MIME type of clear content.
EncodingType	Sub-parameter of tSourceContent. The parameter is optional. When SP issues an RO request for super distribution of the content, the DRM server compares this parameter with the EncodingType parameter of the mobile subscriber. In case of encoding type incompatibility, notification will be sent to SP.
ContentProviderWeb	Sub-parameter of tSourceContent. The parameter is optional. This parameter is used in the DCF header see [1]. The default value of the parameter is configured in the DRM server configuration. The value from configuration is used if the parameter is absent from the RtProtectContentReq request.
ContentVendor	Sub-parameter of tSourceContent. The parameter is optional. This parameter is used in the DCF header see [1]. The default value of the parameter is configured in the DRM server configuration. The value from configuration is used if the parameter is absent in the RtProtectContentReq request.
SourceContentLocation	Sub-parameter of tSourceContent. The parameter is optional. This parameter defines the source content location. If the parameter is absent, the source (clear) content should be sent in the same HTTP stream as the RtProtectContentReq request.

Parameter Name	Description
SourceLocation	Sub-parameter of SourceContentLocation. This parameter defines the URL of the source content location.
SourceDownloadProtocol	Sub-parameter of SourceContentLocation. The parameter is optional. This parameter defines the protocol of the source content download (HTTP, FTP, file system).
SourceDownloadProxy	Sub-parameter of SourceContentLocation. The parameter is optional. The proxy is used for the source content download.
ProtectContentLocation	Sub-parameter of tSourceContent. The parameter is optional. This parameter defines protected content location. If this parameter is absent, the protected content should be sent in the same HTTP stream as the RtProtectContentReqResp message.
ProtLocation	Sub-parameter of ProtectContentLocation. This parameter defines the URL of the protected content location.
ProtUploadProtocol	Sub-parameter of ProtectContentLocation. The parameter is optional. This parameter defines the protocol of the protected content upload (HTTP, FTP, file system).
ProtUploadProxy	Sub-parameter of SourceContentLocation. The parameter is optional. The proxy is used for the protected content upload.
tAssetRights	Complex type, defines asset consumption rights.
Price	Sub-parameter of tAssetRights. Defines the price of the asset. The price is used for statistical purposes only.

Parameter Name	Description
Currency	Sub-parameter of Price. The parameter is optional. This parameter defines the currency is used for the price. The default value of the parameter is configured in the DRM server configuration per SP. The value from configuration is used if the parameter is absent.
Price-value	Sub-parameter of Price. The value that, along with currency, comprises the price.
Permission	Sub-parameter of tAssetRights. The parameter is optional. Permissions as defined in OMA1 (see [2]).
FLflag	Sub-parameter of tAssetRights. Forward Lock flag. The parameter is optional. If the parameter is present, the content must be protected as Separate Delivery with Forward Lock (see [2]).
tmsRights	Complex type defines the rights which can be implemented by the mobile subscriber for the specific MIME type.
Permission	Sub-parameter of tmsRights. Permissions as defined in OMA1 (see [2]).
FLflag	Sub-parameter of tmobile subscriberRights. Forward Lock flag. The parameter is optional. If the parameter present this means that the mobile subscriber of particular type supports Separate Delivery with Forward Lock protection type see [2]).
DRMstatus	Complex parameter. Defines the status of request execution.
ErrorCode	Sub-parameter of DRMstatus.
ErrorDescr	Sub-parameter of DRMstatus. This parameter is used to describe the error code. The parameter is optional.

7 Interface Messages

7.1 Content Protection Transaction

The request message and response message are defined in the upcoming sections.

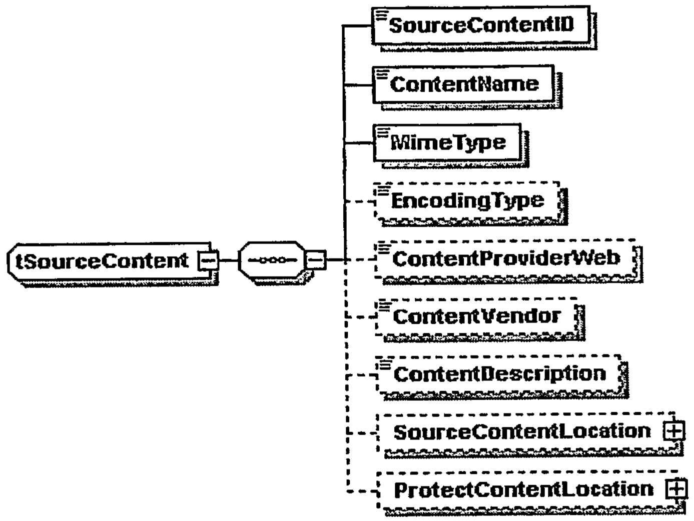
7.1.1 RT Protect Content Request Message

7.1.1.1 Message definition

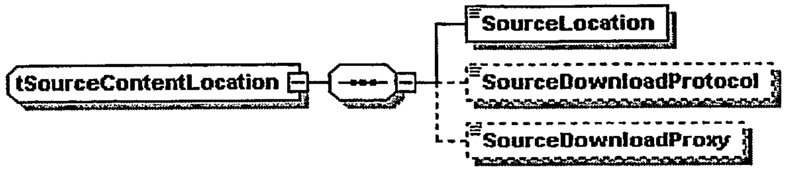
```
<message name="RtProtectContentReqRequest">
        <part name="Version" type="xsd:string"/>
        <part name="srvId" type="xsd:string"/>
        <part name="Source" type="nmdrm:tSourceContent"/>
        <part name="MobSubsId" type="nmdrm:tMobSubsId"/>
        <part name="mobile subscribermodel" type="xsd:string"/>
        <part name="AssetRights" type="nmdrm:tAssetRights"/>
    </message>
```

7.1.1.2 Types used

complexType **tSourceContent**

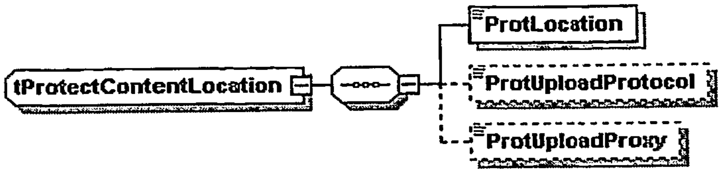
diagram	 SourceContentID ContentName MimeType EncodingType tSourceContent ContentProviderWeb ContentVendor ContentDescription SourceContentLocation ProtectContentLocation
namespace	http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd
children	**SourceContentID ContentName MimeType EncodingType ContentProviderWeb ContentVendor ContentDescription SourceContentLocation ProtectContentLocation**
used by	element **tSourceContent**
source	<xsd:complexType name="tSourceContent"> <xsd:sequence> <xsd:element name="SourceContentID" type="xsd:long"/> <xsd:element name="ContentName" type="xsd:string"/> <xsd:element name="MimeType" type="xsd:string"/> <xsd:element name="EncodingType" type="xsd:string" nillable="true" minOccurs="0"/> <xsd:element name="ContentProviderWeb" type="xsd:string" nillable="true" minOccurs="0"/> <xsd:element name="ContentVendor" type="xsd:string" nillable="true" minOccurs="0"/> <xsd:element name="ContentDescription" type="xsd:string" nillable="true" minOccurs="0"/> <xsd:element name="SourceContentLocation" type="nmdrm:tSourceContentLocation" nillable="true" minOccurs="0"/> <xsd:element name="ProtectContentLocation" type="nmdrm:tProtectContentLocation" nillable="true" minOccurs="0"/> </xsd:sequence> </xsd:complexType>

complexType **tSourceContentLocation**

diagram	 SourceLocation tSourceContentLocation SourceDownloadProtocol SourceDownloadProxy
namespace	http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd
children	**SourceLocation** **SourceDownloadProtocol** **SourceDownloadProxy**
used by	elements **tSourceContent/SourceContentLocation** **tSourceContentLocation**

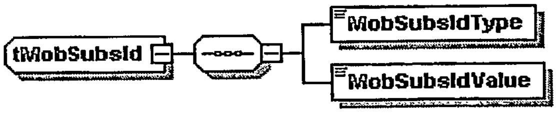
source	<xsd:complexType name="tSourceContentLocation">
	  <xsd:sequence>
	    <xsd:element name="SourceLocation" type="xsd:string"/>
	    <xsd:element name="SourceDownloadProtocol" type="xsd:string" nillable="true" minOccurs="0"/>
	    <xsd:element name="SourceDownloadProxy" type="xsd:string" nillable="true" minOccurs="0"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **tProtectContentLocation**

diagram	 tProtectContentLocation ProtLocation ProtUploadProtocol ProtUploadProxy
namespace	http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd
children	**ProtLocation** **ProtUploadProtocol** **ProtUploadProxy**
used by	elements **tSourceContent/ProtectContentLocation** **tProtectContentLocation**

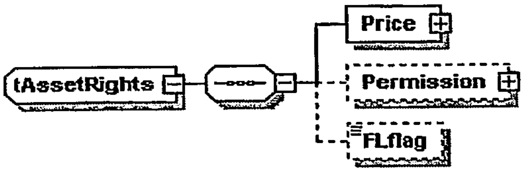
source	<xsd:complexType name="tProtectContentLocation">
	  <xsd:sequence>
	    <xsd:element name="ProtLocation" type="xsd:string"/>
	    <xsd:element name="ProtUploadProtocol" type="xsd:string" nillable="true" minOccurs="0"/>
	    <xsd:element name="ProtUploadProxy" type="xsd:string" nillable="true" minOccurs="0"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **tMobSubsId**

diagram	
namespace	http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd
children	**MobSubsIdType** **MobSubsIdValue**
used by	element **tMobSubsId**

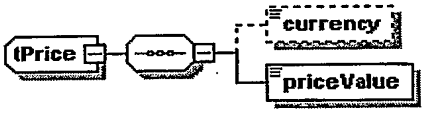
source	<xsd:complexType name="tMobSubsId">
	  <xsd:sequence>
	    <xsd:element name="MobSubsIdType" type="xsd:string"/>
	    <xsd:element name="MobSubsIdValue" type="xsd:string"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **tAssetRights**

diagram	
namespace	http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd
children	**Price** **Permission** **FLflag**
used by	element **tAssetRights**

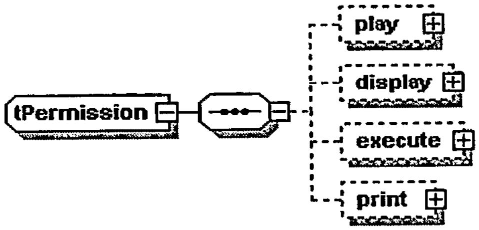
source	<xsd:complexType name="tAssetRights">
	  <xsd:sequence>
	    <xsd:element name="Price" type="nmdrm:tPrice"/>
	    <xsd:element name="Permission" type="nmdrm:tPermission" nillable="true" minOccurs="0"/>
	    <xsd:element name="FLflag" type="xsd:unsignedByte" nillable="true" minOccurs="0"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **tPrice**

diagram	
namespace	http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd
children	**currency** **priceValue**
used by	elements **tAssetRights/Price** **tPrice**

source	<xsd:complexType name="tPrice">
	  <xsd:sequence>
	    <xsd:element name="currency" type="xsd:string" nillable="true" minOccurs="0"/>
	    <xsd:element name="priceValue" type="xsd:float"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **tPermission**

diagram	
namespace	http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd
children	**play** **display** **execute** **print**
used by	elements **tAssetRights/Permission** **tHS-Rights/Permission** **tPermission**

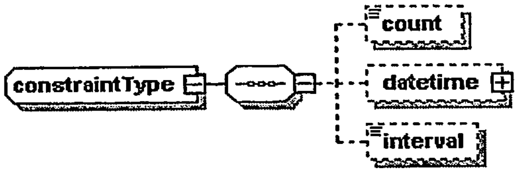
source	<xsd:complexType name="tPermission">
	  <xsd:sequence>
	    <xsd:element name="play" type="nmdrm:tPermissionElement" nillable="true" minOccurs="0"/>
	    <xsd:element name="display" type="nmdrm:tPermissionElement" nillable="true"
	minOccurs="0"/>
	    <xsd:element name="execute" type="nmdrm:tPermissionElement" nillable="true"
	minOccurs="0"/>
	    <xsd:element name="print" type="nmdrm:tPermissionElement" nillable="true" minOccurs="0"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **tPermissionElement**

diagram	tPermissionElement constraint
namespace	http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd
children	**constraint**
used by	elements **tPermission/display** **tPermission/execute** **tPermission/play** **tPermission/print** **tPermissionElement**

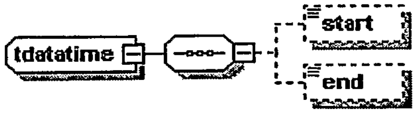
source	<xsd:complexType name="tPermissionElement">
	  <xsd:sequence>
	    <xsd:element name="constraint" type="nmdrm:constraintType"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **constraintType**

diagram	
namespace	http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd
children	**count** **datetime** **interval**
used by	elements **tPermissionElement/constraint** **constraintType**

source	<xsd:complexType name="constraintType">
	  <xsd:sequence>
	    <xsd:element name="count" type="xsd:int" nillable="true" minOccurs="0"/>
	    <xsd:element name="datetime" type="nmdrm:tdatatime" minOccurs="0"/>
	    <xsd:element name="interval" type="xsd:int" nillable="true" minOccurs="0"/>
	  </xsd:sequence>
	</xsd:complexType>

complexType **tdatatime**

diagram	
namespace	http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd
children	**start** **end**
used by	elements **constraintType/datetime** **tdatatime**

source	<xsd:complexType name="tdatatime">
	  <xsd:sequence>
	    <xsd:element name="start" type="xsd:dateTime" nillable="true" minOccurs="0"/>
	    <xsd:element name="end" type="xsd:dateTime" nillable="true" minOccurs="0"/>
	  </xsd:sequence>
	</xsd:complexType>

7.1.1.3 XML example

```
<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:nmdrm="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd">

    <SOAP-ENV:Body id="_0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

        <nmdrm:RtProtectContentReq>

            <Version/>

            <srvId/>

            <Source>

                <SourceContentID>230965</SourceContentID>

                <ContentName "Bond"  />

                <MimeType "application/jpg" />

                <ContentDescription "nice jpg picture"/>

                <SourceContentLocation>

                    <SourceLocation "http://mobile.nds.com/clearcontent"/>

                    <SourceDownloadProxy/>

                </SourceContentLocation>

                <ProtectContentLocation>

                    <ProtLocation "http://mobile.nds.com/protectcontent"/>

                    <ProtUploadProxy/>

                </ProtectContentLocation>

            </Source>

            <MobSubsId>

                <MobSubsIdType "mobile subscriberISDN"/>

                <MobSubsIdValue "97255664541"/>

            </MobSubsId>
            <mobile subscribermodel "Nokia6220" />
            <AssetRights>
                <Price>
                    <priceValue>5.5</priceValue>
                </Price>
                <Permission>
                    <play>
                        <constraint>
                            <count>3</count>
                    </play>
                </Permission>
                <FLflag>0</FLflag>
            </AssetRights>
        </nmdrm:RtProtectContentReq>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

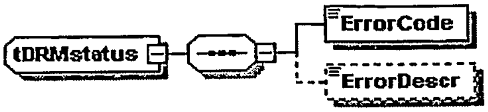
7.1.2 RT Protect Content Response Message

7.1.2.1 Message definition

```
<message name="ProtectContentReqResponse">
        <part name="DRMstatus" type="nmdrm:tDRMstatus"/>
        <part name="CID" type="xsd:string"/>
    </message>
```

7.1.2.2 Types used

complexType **tDRMstatus**

diagram	
namespace	http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd
children	**ErrorCode** **ErrorDescr**
used by	elements **ProtectContentReqResponse/DRMstatus** **RoReqResponse/DRMstatus** **DrmPermissionReqResponse/DRMstatus** **tDRMstatus**

source	<xsd:complexType name="tDRMstatus">
	  <xsd:sequence>
	    <xsd:element name="ErrorCode" type="xsd:int"/>
	    <xsd:element name="ErrorDescr" type="xsd:string" nillable="true" minOccurs="0"/>
	  </xsd:sequence>
	</xsd:complexType>

7.1.2.2.1 XML example

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:nmdrm="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd">
   <SOAP-ENV:Body id="_0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
         <nmdrm:ProtectContentReqResponse>
            <DRMstatus>
               <ErrorCode>0</ErrorCode>
            </DRMstatus>
            <CID "Bondnnnn@mobile.nds.com" />
         </nmdrm:ProtectContentReqResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

7.2 RO Transaction

7.2.1 RO Request Message

7.2.1.1 Message definition

```
<message name="RoReqRequest">
        <part name="Version" type="xsd:string"/>
        <part name="srvId" type="xsd:string"/>
        <part name="CID" type="xsd:string"/>
        <part name="MobSubsId" type="nmdrm:tMobSubsId"/>
        <part name="mobile subscribermodel" type="xsd:string"/>
        <part name="AssetRights" type="nmdrm:tAssetRights"/>
    </message>
```

7.2.1.2 Types used

The same types as for RT Protect Content request are used

7.2.1.3 XML example

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:nmdrm="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd">
    <SOAP-ENV:Body id="_0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
        <nmdrm:RoReq>
            <Version "1.0" />
            <srvId "NDS0124"/>
            <CID "Bondnnnn@mobile.nds.com" />
            <MobSubsId>
                <MobSubsIdType "mobile subscriberISDN"/>
                <MobSubsIdValue "97255664541"/>
            </MobSubsId>
            <mobile subscribermodel "Nokia6220" />
            <AssetRights>
                <Price>
                    <priceValue>7.5</priceValue>
                </Price>
                <Permission>
                    <play>
                        <constraint>
                            <count>5</count>
                    </play>
                </Permission>
                <FLflag>0</FLflag>
            </AssetRights>
        </nmdrm:RoReq>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

7.2.2 RO Response Message

7.2.2.1 Message definition

```
<message name="RoReqResponse">
        <part name="DRMstatus" type="nmdrm:tDRMstatus"/>
        <part name="RightsObject" type="xsd:hexBinary"/>
</message>
```

7.2.2.2 Types used

The same type as for RT Protect Content response is used.

If ErrorCode parameter in the DRMstatus does not equal "OK", the RightsObject parameter in the response message is absent.

7.2.2.3 XML example

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:nmdrm="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd">
    <SOAP-ENV:Body id="_0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
        <nmdrm:RoReqResponse>
            <DRMstatus>
                <ErrorCode>0</ErrorCode>
                <ErrorDescr/>
            </DRMstatus>
            <RightsObject>
                <ptr>0</ptr>
                <size>0</size>
            </RightsObject>
        </nmdrm:RoReqResponse>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

7.3 DRM Permission Transaction

7.3.1 DRM Permission Request Message

7.3.1.1 Message definition

```
<message name="DrmPermissionReqRequest">
        <part name="Version" type="xsd:string"/>
        <part name="srvId" type="xsd:string"/>
        <part name="mobile subscribermodel" type="xsd:string"/>
        <part name="MimeType" type="xsd:string"/>
</message>
```

7.3.1.2 Types used

The same types as in RT request message are used.

7.3.1.3 XML example

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:nmdrm="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd">
    <SOAP-ENV:Body id="_0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
        <nmdrm:DrmPermissionReq>
            <Version> 1.0 </Version>
            <srvId>NDS2300 </srvId>
            <mobile subscribermodel>  Nokia6220   </mobile subscribermodel>
            <MimeType> application/jpg </MimeType>
        </nmdrm:DrmPermissionReq>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

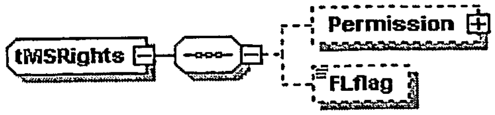
7.3.2 DRM Permission Response Message

7.3.2.1 Message definition

```
<message name="DrmPermissionReqResponse">
        <part name="DRMstatus" type="nmdrm:tDRMstatus"/>
        <part name="mobile subscriberRights" type="nmdrm:tmobile subscriberRights"/>
    </message>
```

7.3.2.2 Types used

complexType **tmobile subscriberRights**

diagram	
namespace	http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd
children	**Permission** **FLflag**
used by	elements **DrmPermissionReqResponse/mobile subscriberRights** **tmobile subscriberRights**

source	`<xsd:complexType name="tmobile subscriberRights">`
	`<xsd:sequence>`
	`<xsd:element name="Permission" type="nmdrm:tPermission" nillable="true" minOccurs="0"/>`
	`<xsd:element name="FLflag" type="xsd:unsignedByte" nillable="true" minOccurs="0"/>`
	`</xsd:sequence>`
	`</xsd:complexType>`

The other types are the same as in the RT Response message.

7.3.2.3 XML example

```
<?xml version="1.0" encoding="UTF-8"?>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:nmdrm="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd">

    <SOAP-ENV:Body id="_0" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

        <nmdrm:DRM-permission-Response>

            <DRM-status>

                <ErrorCode>0</ErrorCode>

                <ErrorDescr/>

            </DRM-status>
```

```
<mobile subscriberRights>
    <Permission>
        <play>
            <constraint>
                <count>0</count>
                <datetime>
                    <start/>
                    <end/>
                </datetime>
                <interval>0</interval>
            </constraint>
        </play>
        <display>
            <constraint>
                <count>0</count>
                <datetime>
                    <start/>
                    <end/>
                </datetime>
                <interval>0</interval>
            </constraint>
        </display>
        <execute>
            <constraint>
                <count>0</count>
                <datetime>
                    <start/>
                    <end/>
                </datetime>
```

```
              <interval>0</interval>
            </constraint>
          </execute>
          <print>
            <constraint>
              <count>0</count>
              <datetime>
                <start/>
                <end/>
              </datetime>
              <interval>0</interval>
            </constraint>
          </print>
        </Permission>
        <FL-flag>0</FL-flag>
      </mobile subscriberRights>
    </nmdrm:DRM-permission-Response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

8 Error Handling

Table 2 below describes the errors that the SP may receive in response to a request.

Table 2: Error Handling

Error code	Error description
0x0000	Successful execution of the request
0x0001	Mobile Subscriber type is not defined in the DRM server
0x0002	Clear content file cannot be found
0x0003	Protected content location does not exist
0x0004	Download protocol is not supported
0x0005	Upload protocol is not supported

Appendix A: WSDL Schema of Protocol

```
<?xml version="1.0" encoding="UTF-8"?>

<definitions name="NDSdrmPortal" xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:SOAP="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:WSDL="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://www.nds.com/NDSDRMS/NDSdrmPortal/NDSdrmPortal.wsdl
" xmlns:tns="http://www.nds.com/NDSDRMS/NDSdrmPortal/NDSdrmPortal.wsdl"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:nmdrm="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd">

    <types>

        <schema targetNamespace="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:nmdrm="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="unqualified"
attributeFormDefault="unqualified">

            <element name="tProtectContentLocation"
type="nmdrm:tProtectContentLocation"/>

            <complexType name="tProtectContentLocation">

                <sequence>

                    <element name="ProtLocation" type="xsd:string" minOccurs="1"
maxOccurs="1"/>

                    <element name="ProtUploadProtocol" type="xsd:string"
minOccurs="0" maxOccurs="1" nillable="true"/>

                    <element name="ProtUploadProxy" type="xsd:string" minOccurs="0"
maxOccurs="1" nillable="true"/>

                </sequence>

            </complexType>

            <element name="tSourceContentLocation"
type="nmdrm:tSourceContentLocation"/>
```

```
        <complexType name="tSourceContentLocation">
            <sequence>
                <element name="SourceLocation" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
                <element name="SourceDownloadProtocol" type="xsd:string"
minOccurs="0" maxOccurs="1" nillable="true"/>
                <element name="SourceDownloadProxy" type="xsd:string"
minOccurs="0" maxOccurs="1" nillable="true"/>
            </sequence>
        </complexType>
        <element name="tSourceContent" type="nmdrm:tSourceContent"/>
        <complexType name="tSourceContent">
            <sequence>
                <element name="SourceContentID" type="xsd:long" minOccurs="1"
maxOccurs="1"/>
                <element name="ContentName" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
                <element name="MimeType" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
                <element name="EncodingType" type="xsd:string" minOccurs="0"
maxOccurs="1" nillable="true"/>
                <element name="ContentProviderWeb" type="xsd:string"
minOccurs="0" maxOccurs="1" nillable="true"/>
                <element name="ContentVendor" type="xsd:string" minOccurs="0"
maxOccurs="1" nillable="true"/>
                <element name="ContentDescription" type="xsd:string"
minOccurs="0" maxOccurs="1" nillable="true"/>
                <element name="SourceContentLocation"
type="nmdrm:tSourceContentLocation" minOccurs="0" maxOccurs="1" nillable="true"/>
                <element name="ProtectContentLocation"
type="nmdrm:tProtectContentLocation" minOccurs="0" maxOccurs="1" nillable="true"/>
            </sequence>
        </complexType>
```

```
            <element name="tMobSubsId" type="nmdrm:tMobSubsId"/>
            <complexType name="tMobSubsId">
                <sequence>
                    <element name="MobSubsIdType" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
                    <element name="MobSubsIdValue" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
                </sequence>
            </complexType>
            <element name="tdatatime" type="nmdrm:tdatatime"/>
            <complexType name="tdatatime">
                <sequence>
                    <element name="start" type="xsd:dateTime" minOccurs="0"
maxOccurs="1" nillable="true"/>
                    <element name="end" type="xsd:dateTime" minOccurs="0"
maxOccurs="1" nillable="true"/>
                </sequence>
            </complexType>
            <element name="constraintType" type="nmdrm:constraintType"/>
            <complexType name="constraintType">
                <sequence>
                    <element name="count" type="xsd:int" minOccurs="0"
maxOccurs="1" nillable="true"/>
                    <element name="datetime" type="nmdrm:tdatatime" minOccurs="0"
maxOccurs="1" nillable="true"/>
                    <element name="interval" type="xsd:int" minOccurs="0"
maxOccurs="1" nillable="true"/>
                </sequence>
            </complexType>
            <element name="tPermissionElement" type="nmdrm:tPermissionElement"/>
            <complexType name="tPermissionElement">
```

```
                <sequence>
                        <element name="constraint" type="nmdrm:constraintType"
minOccurs="1" maxOccurs="1"/>
                </sequence>
        </complexType>
        <element name="tPermission" type="nmdrm:tPermission"/>
        <complexType name="tPermission">
                <sequence>
                        <element name="play" type="nmdrm:tPermissionElement"
minOccurs="0" maxOccurs="1" nillable="true"/>
                        <element name="display" type="nmdrm:tPermissionElement"
minOccurs="0" maxOccurs="1" nillable="true"/>
                        <element name="execute" type="nmdrm:tPermissionElement"
minOccurs="0" maxOccurs="1" nillable="true"/>
                        <element name="print" type="nmdrm:tPermissionElement"
minOccurs="0" maxOccurs="1" nillable="true"/>
                </sequence>
        </complexType>
        <element name="tPrice" type="nmdrm:tPrice"/>
        <complexType name="tPrice">
                <sequence>
                        <element name="currency" type="xsd:string" minOccurs="0"
maxOccurs="1" nillable="true"/>
                        <element name="priceValue" type="xsd:float" minOccurs="1"
maxOccurs="1"/>
                </sequence>
        </complexType>
        <element name="tAssetRights" type="nmdrm:tAssetRights"/>
        <complexType name="tAssetRights">
                <sequence>
```

```
                    <element name="Price" type="nmdrm:tPrice" minOccurs="1"
maxOccurs="1"/>
                    <element name="Permission" type="nmdrm:tPermission"
minOccurs="0" maxOccurs="1" nillable="true"/>
                    <element name="FLflag" type="xsd:unsignedByte" minOccurs="0"
maxOccurs="1" nillable="true"/>
                </sequence>
            </complexType>
            <element name="tmobile subscriberRights" type="nmdrm:tmobile
subscriberRights"/>
            <complexType name="tmobile subscriberRights">
                <sequence>
                    <element name="Permission" type="nmdrm:tPermission"
minOccurs="0" maxOccurs="1" nillable="true"/>
                    <element name="FLflag" type="xsd:unsignedByte" minOccurs="0"
maxOccurs="1" nillable="true"/>
                </sequence>
            </complexType>
            <element name="tDRMstatus" type="nmdrm:tDRMstatus"/>
            <complexType name="tDRMstatus">
                <sequence>
                    <element name="ErrorCode" type="xsd:int" minOccurs="1"
maxOccurs="1"/>
                    <element name="ErrorDescr" type="xsd:string" minOccurs="0"
maxOccurs="1" nillable="true"/>
                </sequence>
            </complexType>
            <element name="ProtectContentReqResponse"
type="nmdrm:ProtectContentReqResponse"/>
            <complexType name="ProtectContentReqResponse">
                <sequence>
```

```
                    <element name="DRMstatus" type="nmdrm:tDRMstatus"
minOccurs="1" maxOccurs="1"/>
                    <element name="CID" type="xsd:string" minOccurs="1"
maxOccurs="1"/>
                </sequence>
            </complexType>
            <element name="RoReqResponse" type="nmdrm:RoReqResponse"/>
            <complexType name="RoReqResponse">
                <sequence>
                    <element name="DRMstatus" type="nmdrm:tDRMstatus"
minOccurs="1" maxOccurs="1"/>
                    <element name="RightsObject" type="xsd:hexBinary" minOccurs="1"
maxOccurs="1"/>
                </sequence>
            </complexType>
            <element name="DrmPermissionReqResponse"
type="nmdrm:DrmPermissionReqResponse"/>
            <complexType name="DrmPermissionReqResponse">
                <sequence>
                    <element name="DRMstatus" type="nmdrm:tDRMstatus"
minOccurs="1" maxOccurs="1"/>
                    <element name="mobile subscriberRights" type="nmdrm:tmobile
subscriberRights" minOccurs="1" maxOccurs="1"/>
                </sequence>
            </complexType>
        </schema>
    </types>
    <message name="RtProtectContentReqRequest">
        <part name="Version" type="xsd:string"/>
        <part name="srvId" type="xsd:string"/>
        <part name="Source" type="nmdrm:tSourceContent"/>
    <part name="MobSubsId" type="nmdrm:tMobSubsId"/>
    <part name="mobile subscribermodel" type="xsd:string"/>
    <part name="AssetRights" type="nmdrm:tAssetRights"/>
</message>
<message name="ProtectContentReqResponse">
    <part name="DRMstatus" type="nmdrm:tDRMstatus"/>
    <part name="CID" type="xsd:string"/>
</message>
<message name="RoReqRequest">
    <part name="Version" type="xsd:string"/>
    <part name="srvId" type="xsd:string"/>
    <part name="CID" type="xsd:string"/>
    <part name="MobSubsId" type="nmdrm:tMobSubsId"/>
    <part name="mobile subscribermodel" type="xsd:string"/>
    <part name="AssetRights" type="nmdrm:tAssetRights"/>
</message>
<message name="RoReqResponse">
    <part name="DRMstatus" type="nmdrm:tDRMstatus"/>
    <part name="RightsObject" type="xsd:hexBinary"/>
</message>
<message name="DrmPermissionReqRequest">
    <part name="Version" type="xsd:string"/>
    <part name="srvId" type="xsd:string"/>
    <part name="mobile subscribermodel" type="xsd:string"/>
    <part name="MimeType" type="xsd:string"/>
</message>
<message name="DrmPermissionReqResponse">
    <part name="DRMstatus" type="nmdrm:tDRMstatus"/>
    <part name="mobile subscriberRights" type="nmdrm:tmobile subscriberRights"/>
    </message>
    <portType name="NDSdrmPortalPortType">
        <operation name="RtProtectContentReq">
            <documentation>Service definition of function
nmdrm__RtProtectContentReq</documentation>
            <input message="tns:RtProtectContentReqRequest"/>
            <output message="tns:ProtectContentReqResponse"/>
        </operation>
        <operation name="RoReq">
            <documentation>Service definition of function
nmdrm__RoReq</documentation>
            <input message="tns:RoReqRequest"/>
            <output message="tns:RoReqResponse"/>
        </operation>
        <operation name="DrmPermissionReq">
            <documentation>Service definition of function
nmdrm__DrmPermissionReq</documentation>
            <input message="tns:DrmPermissionReqRequest"/>
            <output message="tns:DrmPermissionReqResponse"/>
        </operation>
    </portType>
    <binding name="NDSdrmPortalBinding" type="tns:NDSdrmPortalPortType">
        <SOAP:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="RtProtectContentReq">
            <SOAP:operation soapAction=""/>
            <input>
                <SOAP:body use="encoded"
namespace="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
            </input>
```

```
          <output>
               <SOAP:body use="encoded"
namespace="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
          </output>
     </operation>
     <operation name="RoReq">
          <SOAP:operation soapAction=""/>
          <input>
               <SOAP:body use="encoded"
namespace="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
          </input>
          <output>
               <SOAP:body use="encoded"
namespace="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
          </output>
     </operation>
     <operation name="DrmPermissionReq">
          <SOAP:operation soapAction=""/>
          <input>
               <SOAP:body use="encoded"
namespace="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
          </input>
          <output>
               <SOAP:body use="encoded"
namespace="http://www.nds.com/NDSDRMS/v1.0/nmdrm.xsd"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
          </output>
     </operation>
```

```
    </binding>
    <service name="NDSdrmPortal">
        <documentation>gSOAP 2.3 rev 2 generated service definition</documentation>
        <port name="NDSdrmPortal" binding="tns:NDSdrmPortalBinding">
            <SOAP:address
location="http://www.nds.com/NDSDRMS/NDSdrmPortal"/>
        </port>
    </service>
</definitions>
```

Technical Glossary: Acronyms and Abbreviations

Knowledge of the acronyms and abbreviations defined in this technical glossary may be helpful to understanding the information in the present document.

Acronym/Abbreviation	Definition
DCF	DRM Content Format
DRM	Digital Rights Management
DSIG	Digital Signature
HS	Handset
ICD	Interface Component Document
MIME	Multipurpose Internet Mail Extension
MGMS	Management Station
MS	Mobile Subscriber
MSISDN	Mobile Station Integrated Services Digital Network
OMA	Open Mobile Alliance organization
RO	Rights Object
SP	Service Provider
WBXML	WAP (Wireless Application Protocol) Binary Extended Markup Language
WSDL	Web Service Definition Language
XID	Exchange Identifier